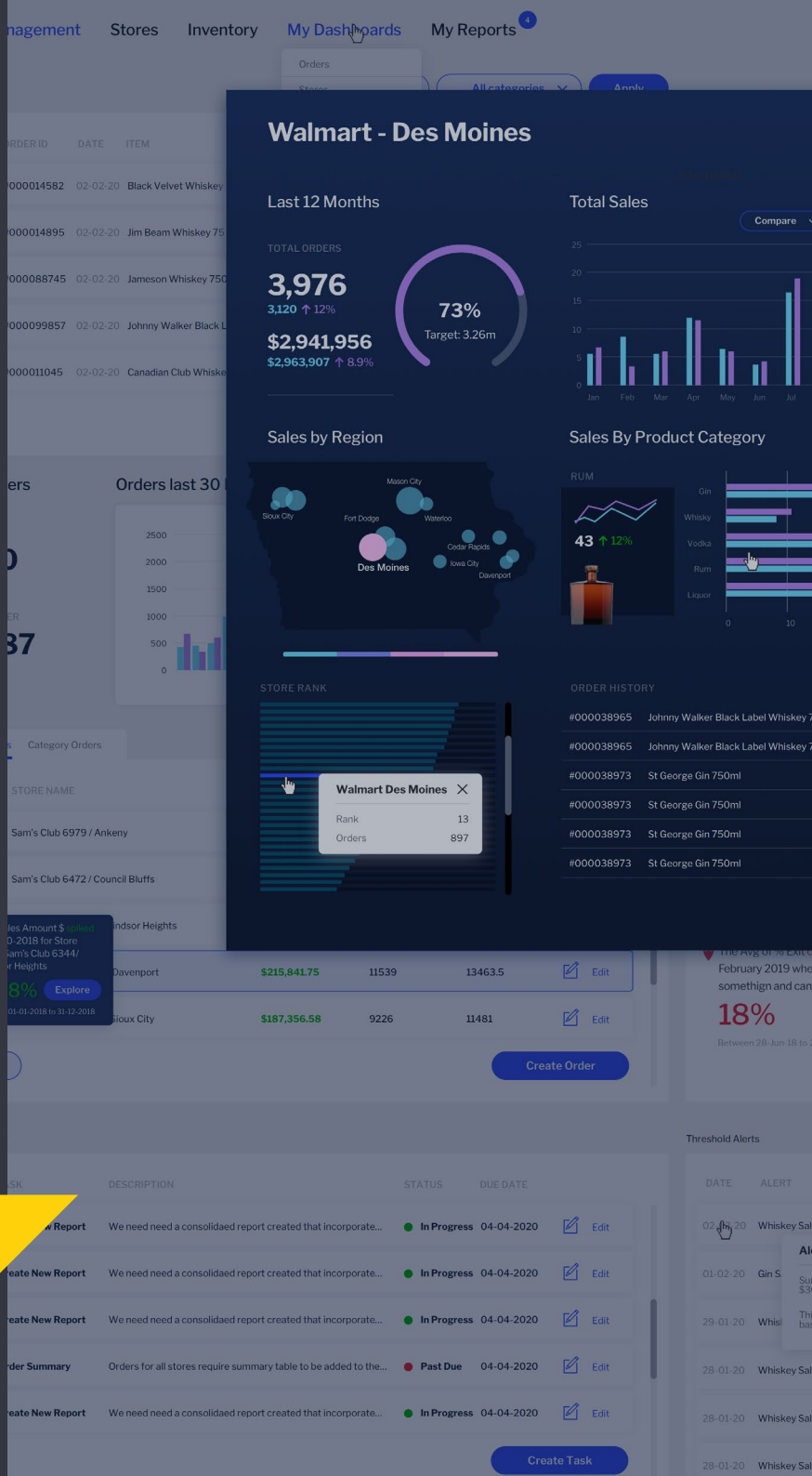
What is Contextual Analytics?

White Paper

November 2020



Today, embedded analytics has made dashboards a core part of most enterprise software. It's enabled everyone to track critical data, monitor performance and explore the metrics they need from their use of your application. However, what embedded analytics can't always easily offer on its own is the deeper context they need to support action and decision-making.

As a product feature for your app, embedded analytics is undoubtedly a valuable tool for all.

But historically, many product managers and developers have approached its integration as a standalone capability. Dashboards and reporting are added as bolted-on functions, rather than as founding strategic components of the core software. Essentially, apps aren't built around data first.

Because analytics is provided as a separate tool, users have to stop what they're doing in their software and switch contexts to access and make effective use of their application's analytics.

To be clear, this approach of embedded analytics is still very valuable. Your users can still seek and get the context needed to support their decision-making - it's just not necessarily an easy process for them when analytics tools are separated from the rest of what they do in your app.

This defining separation of core functions and analytics has led to some long-term limitations:

1. Users must switch contexts or navigate to a separate dashboard module to find the information they need to act, breaking up and being disruptive to their workflow.
2. Stand-alone dashboards do not provide context or guided analysis; instead, users have to manually look for changes or trends or anomalies in their data, which is time consuming and subject to human error and data fatigue.
3. Embedding a separate dashboard module creates some friction in the user experience, and does not guarantee that all users will use it or gain value from it.

Thankfully, in the last few years, embedded analytics solutions have evolved significantly - from simply plugging in a dashboard module, to incorporating contextualized analytics that support and trigger user workflows from within an application - to eliminate these many limitations.

The business benefits of contextual analytics

- Highly differentiated analytical user experiences
- Improved business benefits from your software
- Increased revenue opportunities
- Optimized customer experience with your app
- Significant competitive advantage

In this whitepaper, we explain what contextual analytics is, what it can be used for, and how it can add a huge amount of value to your software solution by improving your user's experience and increasing the business benefits your customers derive from using your application.

What is contextual analytics?

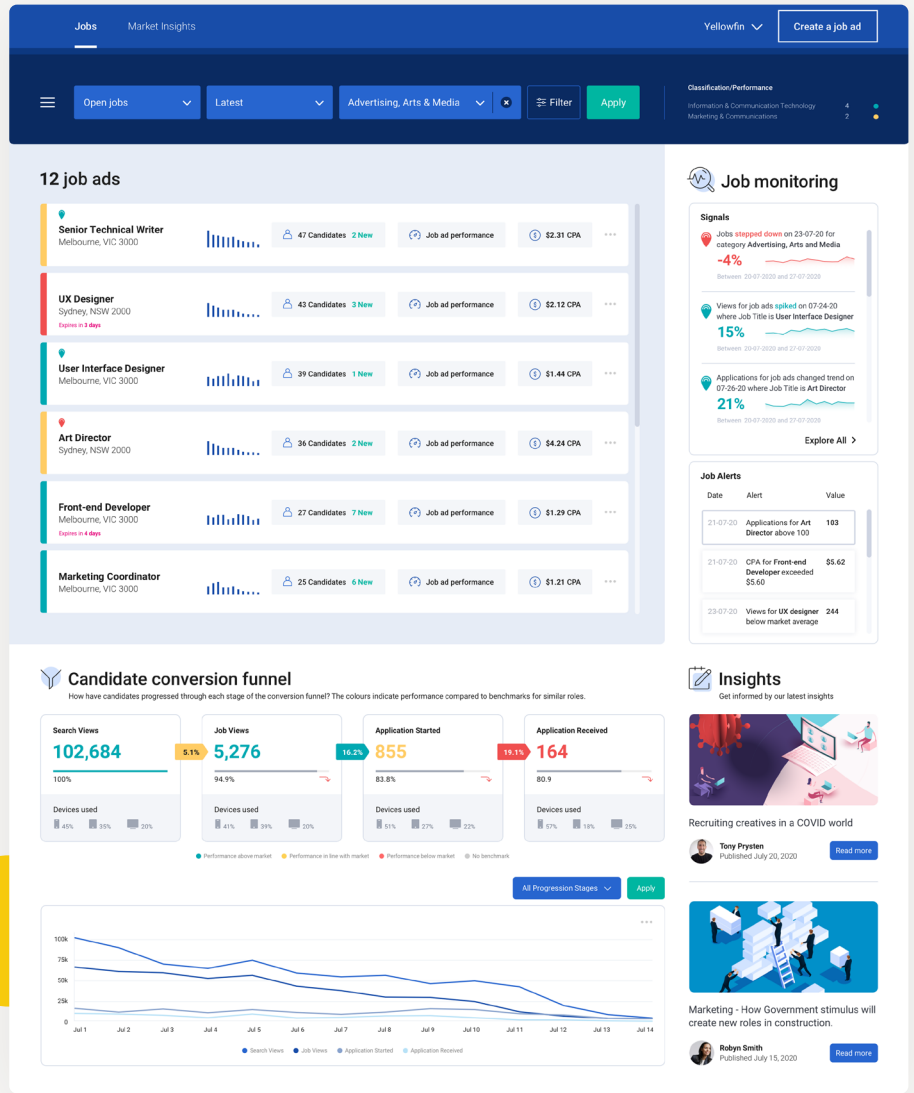
Contextual analytics is when dashboards and reporting solutions are embedded directly into an application's core workflows. As the next level of embedded analytics, it delivers these tools to your users by placing them directly in the situational framework where decisions are made.

In contrast to the traditional form of embedded analytics, which integrates dashboards and reporting standalone modules in your app, your users no longer have to switch away from their transactional environment whenever they need to investigate data or derive insight. With contextual analytics, the data needed to assist their decision-making is now delivered at the point of consumption for the end-user, directly within the user interface and transaction flow.

Embedding analytics into your software in this fashion creates significant value for the user. Picture analytic components - charts, tables, metrics - sitting side by side on the screen with your application's transactional components, communicating bidirectionally with them. With one click, a user now gains instant, guided and dynamic insight, as they are continually guided in their decision-making by your application, while they work as normal. This close merger brings about several new opportunities: For instance, a retail store manager can access daily demand forecasts and be presented with the option of creating shift schedules for next week, while an inventory manager can view inventory trends and replenish orders without exiting the screen.

By definition, contextual analytics is more than a basic chart embedded on the page. The data, the visualization and the associated actions all need to work together in order to provide deeper insight and drive action - and this is what the new phase of embedded analytics makes possible.

Contextual analytics ultimately brings about guided insights that seamlessly integrate into your users' daily work. They don't have to go outside of your app's core workflow for the additional information they need to act, as you're augmenting their analysis capability whilst transacting in your software application. This approach empowers everyone to be more effective in their work.



How does contextual analytics work?

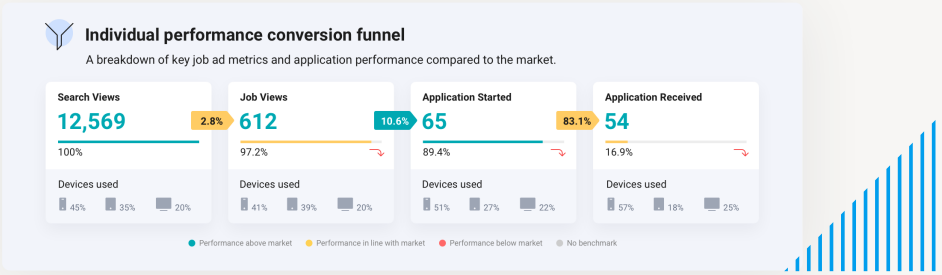
The goal of contextual analytics is to maximize the business benefits that end-users get from their software applications - by either supporting or triggering actions that users make within those applications. To be effective, there are three core criteria that define how contextual analytics needs to work.

1. Visual components are embedded within a software application's pages.
2. Those components are synchronized with the core application.
3. In combination, they support transactional workflows.

Embedded components can take two forms:

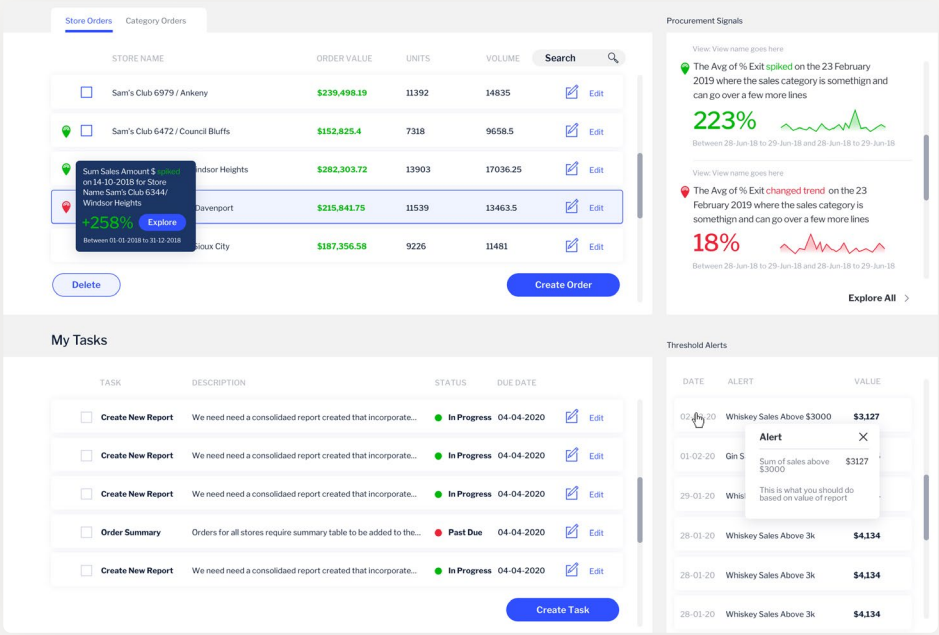
1. **Predefined analytical paths:** Where very simple visualizations, such as KPI numbers, line or bar charts or sophisticated dashboards covering a wide variety of visualizations are presented to the user based on their actions. For example: When viewing an individual customer record, the user can see a series of charts that highlight the history of that customer's purchases. In a predefined path, every customer record will have their own mini dashboard or set of visualizations attached. The visualizations will be the same, but the data will be filtered uniquely for each customer.

   Product managers and owners decide and integrate the most appropriate visualizations to support the end users workflow.



2. **Dynamic analysis and alerts:** Alerts or prompts that indicate either thresholds have been crossed or anomalies in a record have been discovered. In this case, only customers' records that have an alert associated with it will be highlighted; these prompt the user to explore further and take action. The nature of the alert or anomaly will present the user with a dynamic set of visualizations that are consistent with that alert type, but unique for that particular customer and insight.

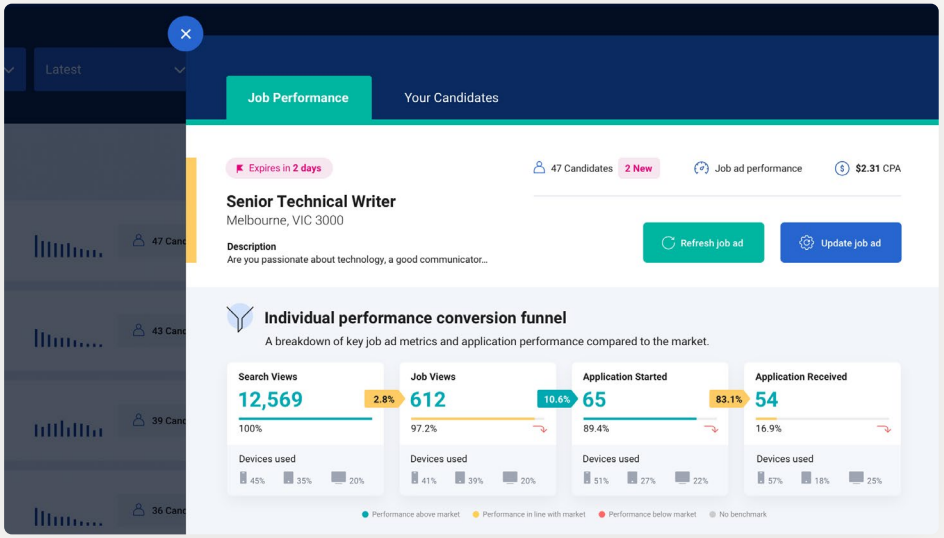
   Product managers and owners decide and integrate the most appropriate alerts needed to support the user - however, automated processes run in the background to identify critical changes and surface these to end users as they occur.



In either case, data being presented could be simple aggregations of transactional data through to the outcome of sophisticated data science models, like propensity to purchase models.

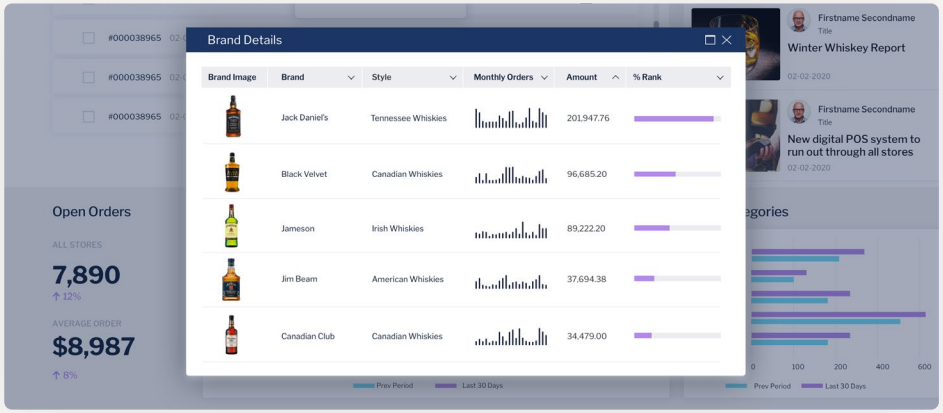
The embedded analytical components must be synchronized with the rest of the page, so that if the context of the page changes, so does the data within the embedded analytic components.

A user may filter the main list of transactions on the page or click on a selected row in a table - these actions result in the analytics being updated to reflect the view that the user is now in. Synchronization is critical if the data presented to the user is to be truly considered contextual.

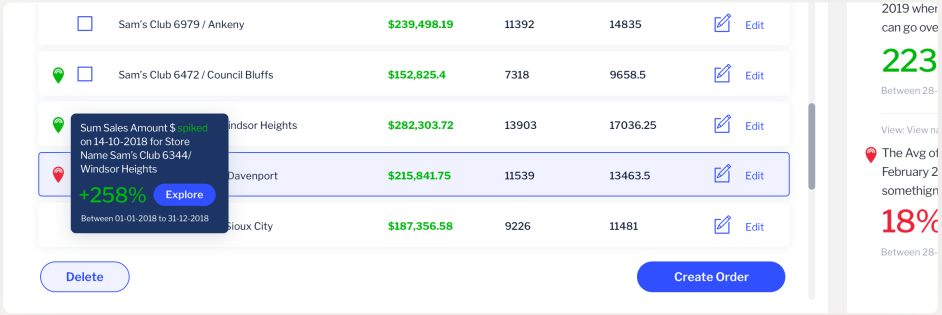


Lastly, contextual analytics should support transactional workflows. Monitoring overall performance and manual data analysis is best done within a standalone dashboard module or individual reports. With contextual analytics, the data presented is either:

1. **Supporting** the current transaction workflow by providing more metrics or insights that specifically pertain to that transaction; or



2. **Trigger** a new workflow via an alert, or suggested actions based on the output of a data science model

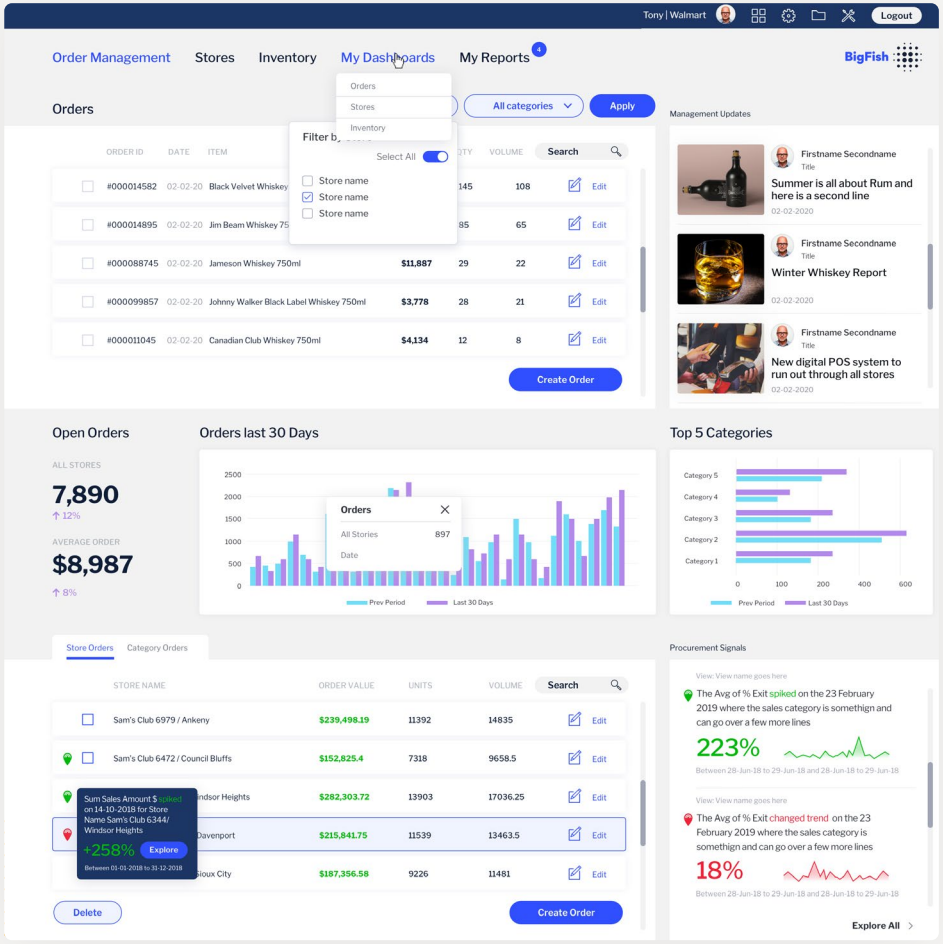


These modes, supporting and triggering are not mutually exclusive and can be combined within a single analytics component. In addition, the workflows or actions can be bi-directional. They can occur in the core app or from within the analytics component, with the primary focus being the simplicity and intuitiveness of the end user experience.

Do you still need dashboards with contextual analytics?

Dashboards will always be important because they enable users to conduct their own independent analysis by slicing and dicing data. Dashboards summarize the most important information at-a-glance, allow users to monitor overall performance, and enable them to drill into the detail behind the numbers. Most critically, they let the user control their own analysis.

Contextual analytics, on the other hand, brings guided analysis to the user. It takes them on an analytics journey, supports their decision-making within a transaction, or acts as the trigger for a decision - that's the key difference, and why you need both capabilities for your embedded BI.



Embedded analytics: Understanding the maturity model

For all the hype surrounding embedded analytics, a remarkable number of software vendors still offer limited reporting or even just basic data extracts for their customers. Low maturity can be the result of limited budgets, lack of skills, inexperience in strategic planning and deployment, or primitive or aging infrastructure.

A good embedded analytics strategy starts with a clear vision. If a software vendor wants to maximize the value of its data assets, they may need to improve maturity levels. To do this, they should focus on their strategy, development capability and tooling first. Low maturity severely constrains vendors who are attempting to modernize their embedded BI. But vendors with low maturity can learn from the success of more mature organizations to speed up modernization of their embedded analytic capabilities, and take their data analytics capabilities to the next level.

The maturity model provides a framework for planning the introduction of embedded analytics capability into a software application. It highlights the value to the end user as well as concentration of analytical development effort on the user or the developer. The model is broken down into five key stages:

Stage 1 - No Capability: Building the initial Minimal Viable Product (MVP) for a new software application and getting it market ready is the primary focus. Reporting capabilities are rarely included in the MVP and the product ships with little to no analytic capability. **Analytic Value = None**.

Stage 2 - Data Exports: Customers start demanding the ability to analyze their data stored in the application; usually a simple CSV download or API access is provided to allow the end user to analyze data in their preferred tool. However, they still need to build all analysis from scratch, leaving a burden of time and expertise on customers. **Analytic Value = Very Low**.

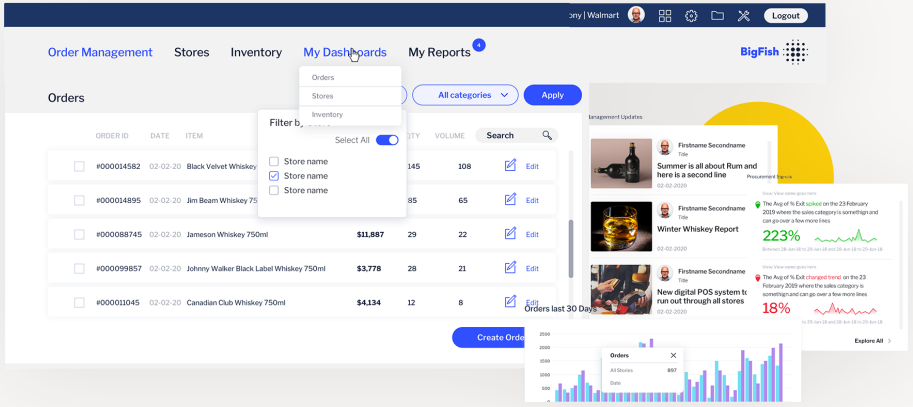
Stage 3 - Basic Reporting: Often the result of an in-house developed solution, users can create basic parameter-driven reports within the product. However, the set of reports or dashboards is extremely limited and users cannot create their own custom analysis. The challenge with this stage is that the customer's need for better insights to drive their decision-making is growing, as are their requests for new reporting capabilities. Developers get bogged down in report creation instead of developing value-added features for the core solution. Most of the analytic workload is shifted away from the customer and onto developers. **Analytic Value = Low**.

Stage 4 - Standalone Dashboard and Reporting Module: Users can perform self-service reporting and analysis with out-of-the-box dashboards and reports and create their own bespoke analytic content using user-friendly tools from pre-defined and secure data sets. As a standalone module users have access to their data and developers are freed from having to create bespoke reports on their behalf. However, this paradigm does not readily guide the user to optimal use of the software. Workflows and decision making is made without the inline support of data. Users need to context switch from the core product workflows to access insights. **Analytic Value = Medium-High**.

Stage 5 - Contextual Analytics: The ability to embed sophisticated automated analysis and in-context reports and dashboards directly into a software application's workflows, guiding users to make optimal use of that software product and improve that product's business benefits. It accomplishes this through the inclusion of blended analytics and bi-directional workflows, with only a minimal workload for your end users and developers. Users derive value as insight discovery is now guided, dynamic and seamless with the overall software application experience. Because a lot of the analytic workload is pre-defined or automated, the effort for both the end user and developer is reduced. **Analytic Value = Very High**.

It's clear that as software developers move through the embedded maturity model, the level of analytics capability provided to end users increases dramatically. The stages are not mutually exclusive but can be additive, especially for stages 4 and 5, so that end users have a variety of mechanisms - from standalone reports and dashboards to contextual analytics - to access the data they need from within their transactional applications.

In the past, many software vendors have lingered in the earlier stages of the maturity curve due to the limitations of older, legacy BI platforms. However, that has changed. Today, elevating the end user analytical experience and giving them real-time information they need to make better decisions is more accessible than it once was - thanks to modern BI vendors that specialise in embedded and contextual analytics.



Contextual Analytics: Why It's Your Next Product Innovation

As organizations become increasingly digital, global and mobile, software vendors need to continuously evolve their embedded analytics offering to ensure their app can proactively provide users with the data, insights and BI capability they need to make informed decisions.

Contextual analytics, embedded into your application's workflows, can be a huge boost to your application's usability and value - and its ability to make insights more relevant is rapidly changing the way leading software vendors choose to embed analytics into their applications.

The ability to embed analytics into a page, synchronise it within the context of the app and enable data to drive workflows has better enabled software developers to create highly differentiated analytical user experiences - providing huge benefits to their end users.

By letting you merge analytics deeper into core workflows, contextual analytics brings about new experiences and workflows that combine insights and actions within your application. Users are guided to optimize the use of the core software application and as a result increase the business benefits derived from that application. As a significant competitive differentiation, contextual analytics will not only improve and create a stickier user experience, reducing churn, but also provide the edge your sales team needs to win more deals.

Above all else, having contextual analytics embedded into your application will increase the business benefit that your users derive from your product. It does this by delivering a better analytical user experience that delights your customers and gives people the critical information they need - when they need it - to interact with your product in a highly optimized way.

Now is the time to prepare for the next era of embedded BI with contextual analytics. By doing so, you highlight the power and the business benefits of your app, and elevate the value of your analytics and user experience. You're helping and giving your customers the power to act with certainty when using your software, while future-proofing your analytics capability for tomorrow.